PATENT

I hereby certify that this correspondence is, on the date shown below, being filed with the U.S. Patent and Trademark Office via EFS.

/Lisa L. Pringle/
Signature
Date: 29 October 2009 Lisa L. Pringle
(type or print name of person certifying)

THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant	:	Randall K. Curey
Serial No.	:	09/821,537
Filing Date	:	28 March 2001
For	:	Partitioned Executive Structure for Real-Time Programs
Group Art Unit	:	2452
Examiner	:	Philip C. Lee
Attorney Docket No.	:	NG(NSD)018807 US CIP

Mail Stop Appeal Briefs - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPEAL BRIEF

Sir:

Pursuant to the Notice of Appeal filed on October 2, 2009, Appellant presents this Appeal Brief.

I. TABLE OF CONTENTS

II. REAL PARTY IN INTEREST

The real party in interest is Northrop Grumman Corporation.

III. RELATED APPEALS AND INTERFERENCES

There are no related appeals or interferences.

IV. STATUS OF CLAIMS

Claims 1-49, which are attached in the first Appendix, are currently pending in this application. Claims 1-2, 4-9, 19-20, 22-27, 29-34, 44-45 and 47-49 stand rejected under 35 U.S.C. §102(b) as being anticipated by U.S. Patent No. 4,109,311 to Blum, et al. ("Blum"). Claims 3 and 28 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of U.S. Patent No. 5,493,649 to Slivka, et al. ("Slivka"). Claims 10-11 and 35-36 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of U.S. Publication No. 2005/0132375 to Douceur, et al. ("Douceur"). Claims 13-17 and 38-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of U.S. Patent No. 5,621,663 to Skagerling ("Skagerling"). Claims 18 and 43 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of Herbert, et al. ("Herbert"). Claims 21 and 46 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of U.S. Patent No. 6,223,201 to Reznak ("Reznak"). Claims 12 and 37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum and Douceur in view of U.S. Patent No. 5,826,092 to Flannery ("Flannery").

The rejection of claims 1-49 is appealed.

V. STATUS OF AMENDMENTS

A response to a Final Office Action ("Final Rejection") issued on August 3, 2009 was filed on September 2, 2009 ("Final Response"). Claims 5 and 20-21 were amended in the Final Response. An Advisory Action Before Filing an Appeal Brief ("Advisory Action") dated September 21, 2009 was issued. The Advisory Action indicated that the amendments to claims 5 and 20-21 had been entered for purposes of Appeal. Moreover, the Advisory Action indicated that the request for reconsideration set forth in the Final Response was considered, but did not place the application in condition for allowance.

VI. SUMMARY OF THE CLAIMED SUBJECT MATTER

A. Claim 1

One aspect of the present invention, as recited in claim 1, is directed to a method for repetitively executing a plurality of software packages at one or more rates, utilizing a common set of computational resources (Page 4, Lines 12-21). A sequence of time intervals is assigned to each software package of the plurality of software packages, the sequence of time intervals assigned to a particular software package of the plurality of software packages not overlapping the sequence of time intervals assigned to any other software package of the plurality of software packages (Page 4, Lines 5-11, Page 10, Lines 18-22). A subset of the plurality of software packages is executed, each respective software package in the subset plurality of software packages being executed during predetermined time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages (Page 4, Lines 12-21, Page 5, Lines 8-10).

B. Claim 21

Claim 21 is directed to the method of claim 1 wherein an executive software package enforces the discipline that each of the respective software packages in the subset of the plurality of software packages is executed only during the time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages, the executive software package determining when the execution of any one of the respective software packages in the subset of the plurality of software packages extends into a time interval defined by the sequence of time intervals assigned to at least one different software package in the subset of the plurality of software packages and performs a remedial action (Page 10, Line 18-Page 11, Line 6).

C. Claim 26

Claim 26 is directed to an apparatus (Page 4, Line 4) for repetitively executing a plurality of software packages at a plurality of rates. The apparatus comprises a means for generating and assigning (Page 4, Line 4; Page 10, line 18) a sequence of time intervals to each software package of the plurality of software packages, the sequence of time intervals assigned to a particular software package of the plurality of software packages not overlapping the sequence of time intervals assigned to any other software package of the plurality of software packages (Page 4, Lines 5-11 and page 10, lines 18-22). The apparatus also comprises a means for executing a subset of the plurality of software packages (Page 4, Line 4, Page 9, Line 15), each respective software package in the subset of the plurality of software packages is executed during predetermined time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages (Page 4, Lines 12-21).

D. Claim 46

Claim 46 is directed to the apparatus of claim 26, wherein an executive software package enforces the discipline that each of the respective software packages in the subset of the plurality of software packages is executed only during time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages, the executive software package determining when the execution of any one of the respective software packages in the subset of the plurality of software packages extends into a time interval defined by the sequence of time intervals assigned to at least one different software package in the subset of the plurality of software packages and performs a remedial action (Page 10, Line 18-Page 11, Line 6).

VII. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

A. Whether claims 1-2, 4-9, 19-20, 22-27, 29-34, 44-45 and 47-49 are anticipated under 35 U.S.C. §102(b) by Blum.

B. Whether claims 3 and 28 are made obvious under 35 U.S.C. §103(a) by Blum in view of Slivka.

C. Whether claims 10-11 and 35-36 are made obvious under 35 U.S.C. §103(a) by Blum in view of Douceur.

D. Whether claims 13-17 and 38-42 are made obvious under 35 U.S.C. §103(a) by Blum in view of Skagerling.

E. Whether claims 18 and 43 are made obvious under 35 U.S.C. §103(a) as being unpatentable over Blum in view of Herbert.

F. Whether claims 21 and 46 are made obvious under 35 U.S.C. §103(a) by Blum in view of Reznak.

G. Whether claims 12 and 37 are made obvious under 35 U.S.C. §103(a) by Blum in view of Douceur and in further view of Flannery.

VIII. ARGUMENT

A. 35 U.S.C. §102(b) rejection of claims 1-2, 4-9, 19-20, 22-27, 29-34, 44-45 and 47-49 as being anticipated by Blum

Anticipation by a single reference requires that the single prior art reference disclose each and every element of the claimed invention, arranged as in the claim. *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 730, F.2d 1452, 1458, 221 U.S.P.Q. 481, 485 (Fed. Cir. 1984).

1. The Anticipation Rejection of Claims 1 and 26

Blum does not disclose that multiple sequences of time are assigned to each software package of a plurality of software packages, and that a subset of the plurality of software packages are executed during predetermined time intervals defined by the assigned sequences of time, as recited in claims 1 and 26.

Blum discloses that allocating time slices to a processing unit can be done two different ways (See Blum, Col. 3, Lines 47-56). In Blum, machine time (time slices) can be divided equally among different programs or assigned dynamically such that a program with a highest priority or a program with the greatest amount of processing time is allocated a greater number of time slices than lower priority programs or programs requiring less processing time (See Blum, Col. 3, Lines 47-53). Moreover, Blum also discloses that if one or two programs are not required for a period of time, computing time (the time slices) can be allocated among the remaining programs (See Blum, Col. 5, Lines 15-19).

Nothing in Blum discloses that a sequence of time intervals are assigned to each software package of a plurality of software packages, and that a subset

of the plurality of software packages are executed during predetermined time intervals defined by the assigned sequences of time, in contrast to claims 1 and 26. Instead, it appears in Blum, programs to be executed are selected "on the fly" (e.g., in real time) during a time slice (e.g., a time interval) based on the needs of the computer programs at the time of the selection. Thus, Blum does not disclose (means for, as recited in claim 26) assigning a sequence of time intervals to each software package of a plurality of software packages, or (means for) executing a subset of a plurality of software packages, as recited in amended claims 1 and 26. In response to the above arguments, in the Final Rejection, the Examiner stated the following:

> [Blum] teaches sequences of time slices in a time cycle are allocated to each program of the plurality of software programs (col. 3, lines 41-44), and that one of the programs (subset) are executed during the allocated time slices (e.g., program 0 is executed during slices 1 and 2, predetermined time intervals) (col. 3, lines 46-54) defined by the allocated sequences of time (defined by the time slices in the time cycle) (Final Rejection, Page 14).

Appellant's representative respectfully submits that the Examiner's interpretation of Blum appears to be contrary to the decision issued by the Board of Patent Appeals and Interferences (BPAI) issued on September 18, 2008 for the present Application. Specifically, page 19 of the BPAI decision states that Blum discloses generating a recurring set of time slice intervals and assigning a time slice to a particular program (emphasis added). The BPAI decision also states that (at the time of the Appeal) in the claims, there is no requirement that a particular sequence of time be assigned to a particular software package. Claims 1 and 26 now explicitly recite (means for) assigning a sequence of time intervals to each software package of a plurality of software packages. Thus, in claims 1 and 26, a particular sequence of time is assigned to a particular software package. In contrast to claims 1 and 26, in Blum, no sequences of time intervals are assigned. Instead, in Blum a sequence of program execution is

defined for a set of time slices (See Blum Col. 3, Lines 40-55). Therefore, Blum does not disclose assigning a sequence of time intervals to each software package of a plurality of software packages, as recited in claims 1 and 26.

For at least the reasons stated above, Blum does not anticipate claims 1 and 26. Accordingly, Appellant's representative respectfully requests that the rejection of claims 1 and 26 be withdrawn.

2. The Obviousness Rejection of Claims 2, 4-9, 19-20, 22-25, 27, 29-34, 44-45 and 47-49

Claims 2, 4-9, 19-20, 22-25, 27, 29-34, 44-45 and 47-49 depend from claims 1 and 268, and are patentable for at least the same reasons as claims 1 and 26, and for the specific elements recited therein. Accordingly, withdrawal of the rejection of claims 2, 4-9, 19-20, 22-25, 27, 29-34, 44-45 and 47-49 is respectfully requested.

B. 35 U.S.C. §103(a) rejection of claims 3 and 28 as being made obvious by Blum in view of Slivka

Claims 3 and 28 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of Slivka. Claims 3 and 8 depend from claims 1 and 26, respectively, and are patentable for at least the same reasons as claims 1 and 26, and for the specific elements recited therein. Moreover, in rejecting claims 3 and 28, the Examiner cites Slivka solely for Slivka's disclosure of a checksum test (See Final Rejection, Page 8, citing Col. 1, Lines 56-67 of Slivka). However, the addition of Slivka does not make up for the aforementioned deficiencies of Blum with respect to claims 1 and 26, from which claims 3 and 28 depend. Accordingly, claims 3 and 28 should be patentable over the cited art and withdrawal of this rejection is respectfully requested.

C. Rejection of Claims 10-11 and 35-36 Under 35 U.S.C. §103(a)

Claims 10-11 and 35-36 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of Douceur. Claims 10-11 and 35-36 depend from claims 1 and 26, and are patentable for at least the same reasons as claims 1 and 26, and for the specific elements recited therein. Moreover, in rejecting claims 10-11 and 35-36, the Examiner cites Douceur solely for Douceur's disclosure of background and foreground processes (See Final Rejection, Pages 9-10, citing Par. [0005] of Douceur). However, the addition of Douceur does not make up for the aforementioned deficiencies of Blum with respect to claims 1 and 26, from which claims 10-11 and 35-36 depend. Accordingly, claims 10-11 and 35-36 should be patentable over the cited art and withdrawal of this rejection is respectfully requested.

D. Rejection of Claims 13-17 and 38-42 Under 35 U.S.C. §103(a)

Claims 13-17 and 38-42 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of U.S. Patent No. 5,621,663 to Skagerling ("Skagerling"). Claims 13-17 and 38-42 depend from claims 1 and 26, and are patentable for at least the same reasons as claims 1 and 26, and for the specific elements recited therein. Moreover, in rejecting claims 13-17 and 38-42, the Examiner cites Skagerling solely for Skagerling's disclosure of a log file (See Final Rejection, Page 9, citing Col. 4, Lines 54-57 of Skagerling). However, the addition of Skagerling does not make up for the aforementioned deficiencies of Blum with respect to claims 1 and 26, from which claims 13-17 and 38-42 depend. Accordingly, claims 13-17 and 38-42 should be patentable over the cited art and withdrawal of this rejection is respectfully requested.

E. Rejection of Claims 18 and 43 Under 35 U.S.C. §103(a)

Claims 18 and 43 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of Herbert. Claims 18 and 43 depend from claims 1 and 26, respectively, and are patentable for at least the same reasons as claims 1 and 26, and for the specific elements recited therein. Moreover, in rejecting claims 18 and 43, the Examiner cites Herbert solely for Herbert's disclosure of an isolated execution ring (See Final Rejection, Page 12, citing Pars. [0021], [0025] and [0041] of Herbert). However, the addition of Herbert does not make up for the aforementioned deficiencies of Blum with respect to claims 1 and 26, from which claims 18 and 43 depend. Accordingly, claims 18 and 43 should be patentable over the cited art and withdrawal of this rejection is respectfully requested.

F. Rejection of Claims 21 and 46 Under 35 U.S.C. §103(a)

Claims 21 and 46 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum in view of Reznak.

Combining and modifying Blum and Reznak in a manner that would make claims 21 and 46 obvious would require a substantial redesign of Blum that would change a basic principle of Blum's operation.

Claims 21 and 46 depend from claims 1 and 26, respectively and recite an executive software package that enforces a discipline that each respective software packages in the subset of the plurality of software packages is executed only during time intervals defined by a sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages. In claims 21 and 46, the executive software package determines when the execution of any one of the respective software packages in the subset of the plurality of software packages extends into a time interval defined by the sequence of time intervals assigned to at least one different software package in

the subset of the plurality of software packages. In rejecting claims 21 and 26, the Examiner admits that Blum fails to disclose that execution of a software package extends into a time interval assigned to another software package, but contends that the teachings of Reznak make up for the deficiencies of Blum (See Final Rejection, Page 15). Appellant's representative respectfully disagrees. In Blum, a particular time slice is not associated with a program until after that program has been selected to be executed in the particular time slice, which selection occurs immediately prior to the execution (See e.g., Blum, Col. 4, Lines 40-56). That is, Blum assigns time slices to programs in real time. Since in Blum, programs are selected to be executed in real time, there is never a time that a program is being executed in a time slice that is associated with a different program.

A claim is not obvious where a suggested combination of references would require a substantial redesign and reconstruction of the elements shown in the prior art as well as a change in the basic principle under which the prior art was designed to operate. *In re Ratti*, 270 F.2d 810, 813, 123 U.S.P.Q. 349 (C.C.P.A. 1959). To modify Blum in such a manner that would make claims 21 and 46 obvious would require the inclusion of additional overhead systems that would have to assign particular time slices (e.g., sequence of time intervals) to programs (e.g., software packages) well before the duration of the time slice. Appellant's representative respectfully submits that the inclusion of such additional overhead would constitute a substantial redesign and reconstruction of Blum. Moreover, Appellant's representative respectfully submits that the purported modifications to Blum would change a basic principle of operation of Blum since the purported combination and modification would require that Blum not select programs to execute in time slices in real time, which appears to be a basic principle of operation of Blum. Accordingly, Appellant's representative respectfully submits that combining and modifying Blum with any other reference

(including Reznak) in a manner that would make claims 21 and 46 obvious would both (1) require a substantial redesign and reconstruction of Blum that would (2) change a basic principle of operation in Blum. Thus, Appellant's representative respectfully submits that it would not have been obvious to one of ordinary skill in the art to combine and modify the teachings of Blum with the teachings of Reznak in a manner that would make claims 21 and 46 obvious.

For at least the reasons stated above, Blum taken in view of Reznak does not make claims 21 and 46 obvious to one of ordinary skill in the art, and claims 21 and 46 should be patentable over the cited art. Thus, withdrawal of this rejection is respectfully requested.

VIII. Rejection of Claims 12 and 37 Under 35 U.S.C. §103(a)

Claims 12 and 37 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Blum and Douceur in view of Flannery. Claims 12 and 37 depend from claims 1 and 26, respectively, and are patentable for at least the same reasons as claims 1 and 26, and for the specific elements recited therein. Moreover, in rejecting claims 12 and 37, the Examiner cites Flannery solely for Flannery's disclosure of a program that conserves power (See Final Rejection, Page 16, citing Col. 3., Lines 51-58 of Flannery). However, the addition of Flannery does not make up for the aforementioned deficiencies of Blum with respect to claims 1 and 26, from which claims 12 and 37 depend. Accordingly, claims 12 and 37 should be patentable over the cited art and withdrawal of this rejection is respectfully requested.

IX. APPENDICES

The first attached Appendix contains a copy of the claims on appeal.

The second and third Appendices have been included to comply with statutory requirements.

Please charge any deficiency or credit any overpayment in the fees for this Appeal Brief to Deposit Account No. 20-0090.

Respectfully submitted,

/Christopher P. Harris/
Christopher P. Harris
Reg. No. 43,660

TAROLLI, SUNDHEIM, COVELL
& TUMMINO, L.L.P.
1300 East Ninth Street, Suite 1700
Cleveland, Ohio 44114
(216) 621-2234
(216) 621-4072 (Facsimile)
Customer No.: 26294

Claims Appendix

Claim 1 (Finally Rejected) A method for repetitively executing a plurality of software packages at one or more rates, utilizing a common set of computational resources, the method comprising the steps:

assigning a sequence of time intervals to each software package of the plurality of software packages, the sequence of time intervals assigned to a particular software package of the plurality of software packages not overlapping the sequence of time intervals assigned to any other software package of the plurality of software packages;

executing a subset of the plurality of software packages, each respective software package in the subset plurality of software packages being executed during predetermined time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages.

Claim 2 (Finally Rejected) The method of claim 1 further comprising the step of utilizing one or more tests to identify the plurality software packages that are valid, and wherein the subset of the plurality of software packages includes only valid software packages.

Claim 3 (Finally Rejected) The method of claim 2 wherein a given test of the one or more tests for validity is a one's complement checksum test of a software package's program memory data.

Claim 4 (Finally Rejected) The method of claim 2 wherein a given software package of the plurality of software packages is assigned a dedicated memory

region, a given test of the one or more tests for validity being whether an address returned for an initialization procedure of the given software package of the plurality of software packages lies within the dedicated memory region of the given software package of the plurality of software packages.

Claim 5 (Finally Rejected) The method of claim 4 wherein the given test of the one or more tests is whether the address is returned for the initialization procedure of the given software package of the plurality of software packages within a predetermined time.

Claim 6 (Finally Rejected) The method of claim 2 wherein a given software package of the plurality of software packages is assigned a dedicated memory region, the dedicated memory region of the given software package of the plurality of software packages including a stack memory region and/or a heap memory region, a given test of the one or more tests for validity being whether the stack memory region and/or the heap memory region assigned during the execution of an initialization procedure of the given software package of the plurality of software packages and various associated entry points lies within the dedicated memory region assigned to the given software package of the plurality of software packages.

Claim 7 (Finally Rejected) The method of claim 6 wherein the given test of the one or more tests is whether the stack memory region and/or the heap memory region and the various associated entry points are returned within a predetermined time.

Claim 8 (Finally Rejected) The method of claim 1 wherein a given software package of the plurality of software packages is assigned a dedicated memory region.

Claim 9 (Finally Rejected) The method of claim 8 wherein the dedicated memory region assigned to the given software package of the plurality of software packages includes a stack memory region in which a stack of the given software package of the plurality of software packages resides.

Claim 10 (Finally Rejected) The method of claim 1 wherein a given software package of the plurality of software packages includes background tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed.

Claim 11 (Finally Rejected) The method of claim 10 wherein a given background task of the background tasks is an infinite loop.

Claim 12 (Finally Rejected) The method of claim 10 wherein the given software package of the plurality of software packages causes power utilized in executing the given software package of the plurality of software packages to be minimized after completion of the background tasks.

Claim 13 (Finally Rejected) The method of claim 1 wherein a failure in the execution of a given software package of the plurality of software packages causes information to be logged in a failure log.

Claim 14 (Finally Rejected) The method of claim 13 wherein a failure in execution is linked to the given software package of the plurality of software packages that caused the failure.

Claim 15 (Finally Rejected) The method of claim 13 wherein quality of performance in executing the given software package of the plurality of software packages is represented by one or more performance-quality parameters, values of the one or more performance-quality parameters being determined from the information logged in the failure log, the execution of the given software package of the plurality of software packages being subject to a plurality of execution options, an execution option being selected on the basis of the values of the one or more performance-quality parameters.

Claim 16. (Finally Rejected) The method of claim 15 wherein the plurality of execution options are user configurable.

Claim 17. (Finally Rejected) The method of claim 15 wherein the one or more performance-quality parameters include the number of failures and/or the rate of failures for one or more classes of failures recorded in the failure log.

Claim 18. (Finally Rejected) The method of claim 1 wherein safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software.

Claim 19. (Finally Rejected) The method of claim 1 wherein each software package of the plurality of software packages is assigned a memory block, a given software package of the plurality of software packages being configured to read data only from zero or more memory blocks associated with other software packages of the plurality of software packages, the zero or more memory blocks readable by the given software package of the plurality of software packages being either predetermined or determined during execution of the given software package of the plurality of software packages in accordance with a set of one or more rules.

Claim 20. (Finally Rejected) The method of claim 1 wherein each software package of the plurality of software packages is assigned a memory block, a given software package of the plurality of software packages being configured to write data only to zero or more memory blocks associated with other software packages of the plurality of software packages, the zero or more memory blocks writeable by the given software package of the plurality of software packages being either predetermined or determined during execution of the given software package of the plurality of software packages in accordance with a set of one or more rules.

Claim 21. (Finally Rejected) The method of claim 1 wherein an executive software package enforces the discipline that each of the respective software packages in the subset of the plurality of software packages is executed only during the time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages, the executive software package determining when the execution of any one of the respective software packages in the subset of the plurality of software packages extends into a time interval defined by the sequence of time intervals assigned to at least one different software package in the subset of the plurality of software packages and performs a remedial action.

Claim 22. (Finally Rejected) The method of claim 1 wherein a presence of the subset of the plurality of software packages is detected.

Claim 23. (Finally Rejected) The method of claim 1 wherein one or more software packages of the plurality of software packages are independently compiled, linked, and loaded.

Claim 24. (Finally Rejected) The method of claim 1 wherein each software package of the plurality of software packages has a stack that is selected prior to executing the software package.

Claim 25. (Finally Rejected) Apparatus for practicing the method of claim 1.

Claim 26. (Finally Rejected) Apparatus for repetitively executing a plurality of software packages at a plurality of rates, the apparatus comprising:

a means for generating and assigning a sequence of time intervals to each software package of the plurality of software packages, the sequence of time intervals assigned to a particular software package of the plurality of software packages not overlapping the sequence of time intervals assigned to any other software package of the plurality of software packages;

a means for executing a subset of the plurality of software packages, each respective software package in the subset of the plurality of software packages is executed during predetermined time intervals defined by the sequence of time intervals assigned to the respective software package in the subset of the plurality of software packages.

Claim 27. (Finally Rejected) The apparatus of claim 26 further comprising a means for utilizing one or more tests to identify the software packages that are valid, and wherein the subset of the plurality of software packages includes only valid software packages.

Claim 28. (Finally Rejected) The apparatus of claim 27 wherein a given test of the one or more tests for validity is a one's complement checksum test of a software package's program memory data.

Claim 29. (Finally Rejected) The apparatus of claim 27 wherein a given software package of the plurality of software packages is assigned a dedicated memory region, a given test of the one or more tests for validity being whether an address returned for an initialization procedure of the given software package of the plurality of software packages lies within the dedicated memory region of the given software package of the plurality of software packages.

Claim 30. (Finally Rejected) The apparatus of claim 29 wherein the given test of the one or more tests is whether the address is returned within a predetermined time.

Claim 31. (Finally Rejected) The apparatus of claim 27 wherein a given software package of the plurality of software packages is assigned a dedicated memory region, the dedicated memory region of the given software package of the plurality of software packages including a stack memory region and/or a heap memory region, a given test of the one or more tests for validity being whether the stack memory region and/or the heap memory region assigned during the

execution of an initialization procedure of the given software package of the plurality of software packages and various associated entry points lies within the dedicated memory region of the given software package of the plurality of software packages.

Claim 32. (Finally Rejected) The apparatus of claim 31 wherein the given test of the one or more tests is whether the stack memory range and/or the heap memory range and the various associated entry points are returned within a predetermined time.

Claim 33. (Finally Rejected) The apparatus of claim 26 wherein a given software package of the plurality of software packages is assigned a dedicated memory region.

Claim 34. (Finally Rejected) The apparatus of claim 33 wherein the dedicated memory region of the given software package of the plurality of software packages includes a stack memory region in which a stack of the given software package of the plurality of software packages resides.

Claim 35. (Finally Rejected) The apparatus of claim 26 wherein a given software package of the plurality of software packages includes background

tasks as well as foreground tasks, the background tasks being performed after the foreground tasks have been completed.

Claim 36. (Finally Rejected) The apparatus of claim 35 wherein a given background task of the background tasks is an infinite loop.

Claim 37. (Finally Rejected) The apparatus of claim 35 wherein the given software package of the plurality of software packages causes power utilized in executing the given software package of the plurality of software packages to be minimized after completion of the background tasks.

Claim 38. (Finally Rejected) The apparatus of claim 26 wherein a failure in the execution of a given software package of the plurality of software packages causes information to be logged in a failure log.

Claim 39. (Finally Rejected) The apparatus of claim 38 wherein a failure in execution is linked to the given software package of the plurality of software packages that caused the failure.

Claim 40. (Finally Rejected) The apparatus of claim 38 wherein quality of performance in executing the given software package of the plurality of software

packages is represented by one or more performance-quality parameters, values of the one or more performance-quality parameters being determined from the information logged in the failure log, the execution of the given software package of the plurality of software packages being subject to a plurality of execution options, an execution option being selected on the basis of the values of the one or more performance-quality parameters.

Claim 41. (Finally Rejected) The apparatus of claim 40 wherein the plurality of execution options are user configurable.

Claim 42. (Finally Rejected) The apparatus of claim 40 wherein the one or more performance-quality parameters include the number of failures and/or the rate of failures for one or more classes of failures recorded in a the failure log.

Claim 43. (Finally Rejected) The apparatus of claim 26 wherein safety-critical software is placed in one or more separate partitions thereby isolating the safety-critical software from non-safety-critical software.

Claim 44. (Finally Rejected) The apparatus of claim 26 wherein each software package of the plurality of software packages is assigned a memory block, a given software package of the plurality of software packages being

configured to read data only from zero or more memory blocks associated with other software packages of the plurality of software packages, the zero or more memory blocks readable by the given software package of the plurality of software packages being either predetermined or determined during execution of the given software package of the plurality of software packages in accordance with a set of one or more rules.

Claim 45. (Finally Rejected) The apparatus of claim 26 wherein each software package of the plurality of software packages is assigned a memory block, a given software package of the plurality of software packages being configured to write data only to zero or more memory blocks associated with other software packages of the plurality of software packages, the zero or more memory blocks writeable by the given software package of the plurality of software packages being either predetermined or determined during execution of the given software package of the plurality of software packages in accordance with a set of one or more rules.

Claim 46. (Finally Rejected) The apparatus of claim 26 wherein an executive software package enforces the discipline that each of the respective software packages in the subset of the plurality of software packages is executed only during time intervals defined by the sequence of time intervals assigned to the

respective software package in the subset of the plurality of software packages, the executive software package determining when the execution of any one of the respective software packages in the subset of the plurality of software packages extends into a time interval defined by the sequence of time intervals assigned to at least one different software package in the subset of the plurality of software packages and performs a remedial action.

Claim 47. (Finally Rejected) The apparatus of claim 26 wherein a presence of the subset of the plurality of software packages is detected.

Claim 48. (Finally Rejected) The apparatus of claim 26 wherein one or more software packages of the plurality of software packages are independently compiled, linked, and loaded.

Claim 49. (Finally Rejected) The apparatus of claim 26 wherein each software package of the plurality of software packages has a stack that is selected prior to executing the software package.

Evidence Appendix

None

Related Proceedings Appendix

None